Apéndice C

Manual de usuario

En esta sección se presentara paso a paso la manera en que se utiliza Voai para generar servidores OAI. Se lleva al usuario de la mano en cada fase de Voai y se le muestra la forma en que debe suministrar información a la aplicación.

C.1. Interfaz principal

Una vez que Voai se ha instalado adecuadamente, se accede a esta aplicación con un navegador de Internet a través de la dirección http://localhost:8080/Voai/servlets.

En esta dirección se muestra la interfaz principal de Voai. Esta interfaz tiene hipervínculos que corresponden a cada una de las fases que hay que cubrir para generar un servidor OAI. Esta interfaz se muestra en la figura C.1.

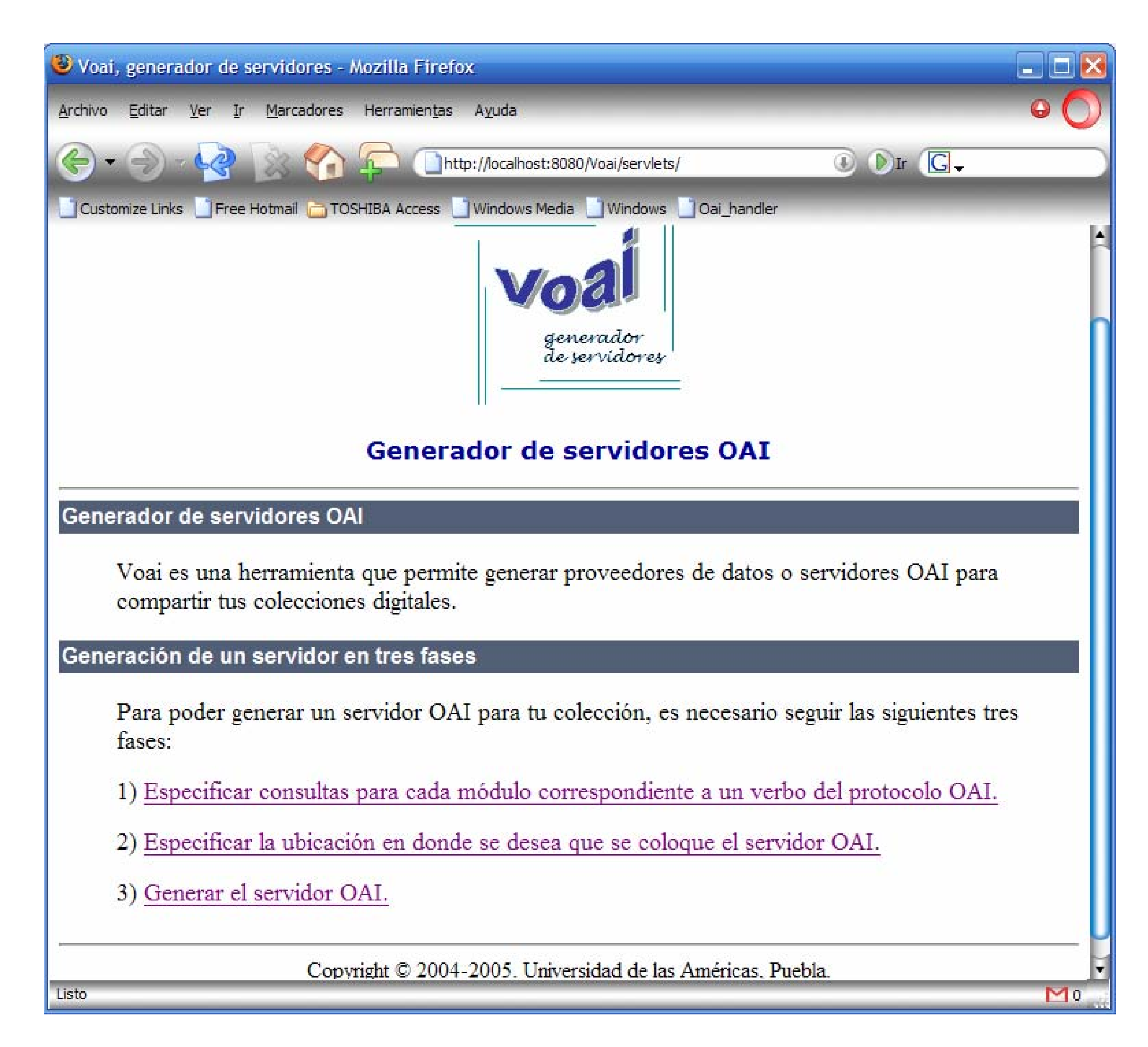


Figura C.1. Interfaz principal de Voai.

C.2. Fase uno: suministro de información.

Cuando se selecciona el primer hipervínculo de la interfaz principal, es decir, el hipervínculo referente a la especificación de consultas para cada verbo del protocolo, se despliega la interfaz de la figura C.2.

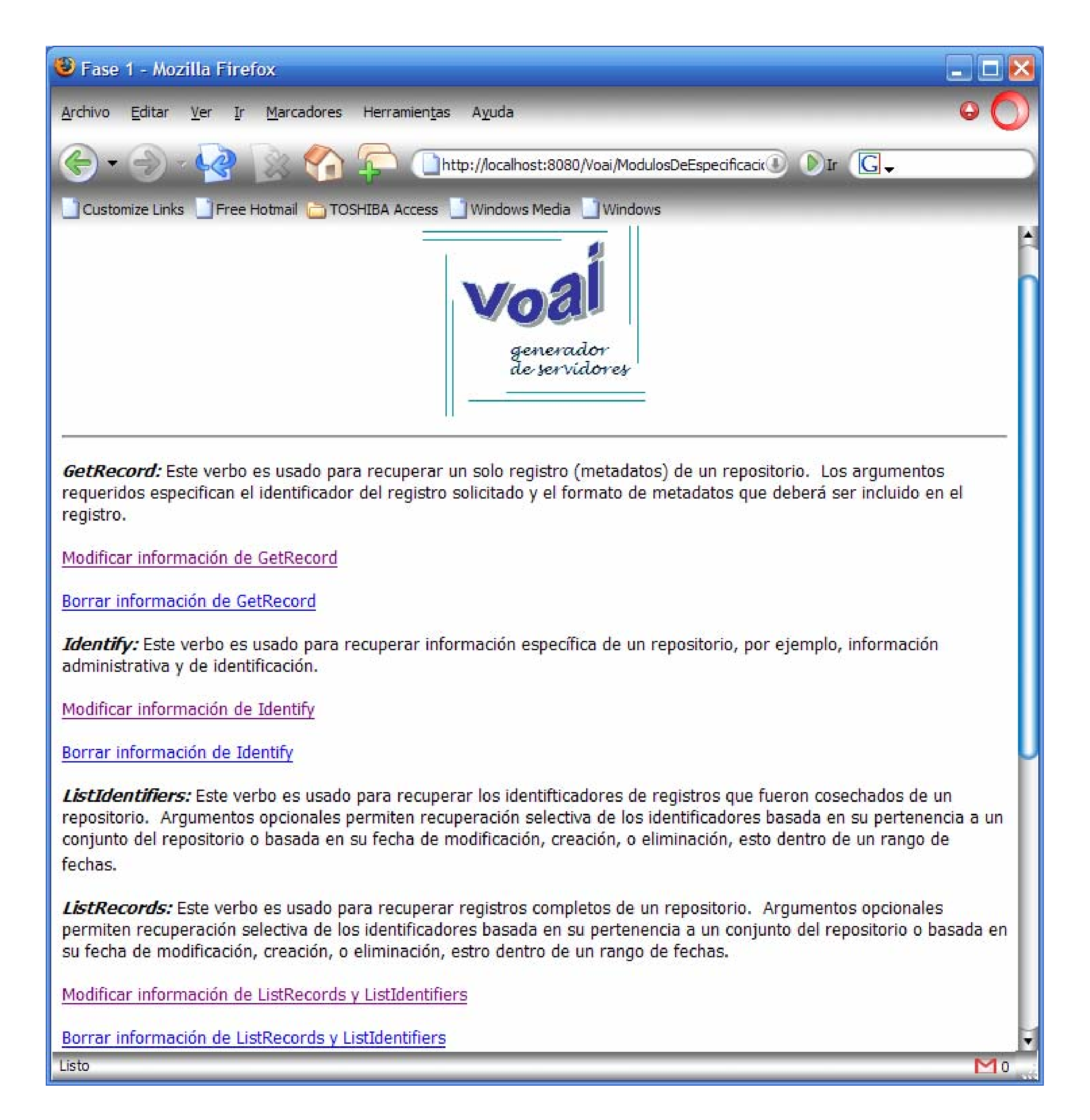


Figura C.2. Interfaz de la fase 1.

En Voai, a cada tipo de petición OAI-PMH le corresponde un módulo de especificación. Esta correspondencia uno a uno se aplica para todas las peticiones excepto para ListRecords y ListIdentifiers. Debido a que la forma de responder a estos dos tipos de peticiones es muy similar, fue posible generalizar la información necesaria para ambas peticiones en un solo módulo de especificación. Entonces existen cinco módulos de

especificación en total, uno correspondiente a ListRecords y ListIdentifiers, y los otros cuatro correspondientes a los cuatro verbos restantes del protocolo.

Como se puede ver, para cada módulo de especificación existen dos hipervínculos: uno que permite suministrar información correspondiente a esa petición o bien modificarla, y el otro permite borrar completamente la información previamente proporcionada. A continuación se explica a detalle la forma en que se proporciona información a cada módulo, pero para eso, antes es necesario ver la sintaxis en que se proporciona esa información.

C.2.1 Sintaxis de información

Existen tres tipos de información que se suministra a los módulos de especificación:

- **Información de identificación**

  Este tipo de información se suministra solamente en el módulo Identify. Esta información no tiene una sintaxis especial, es decir, se proporciona la información de manera natural.

- **Información de conexión a la base de datos**

  Este tipo de información se suministra solamente en el módulo ListSets. Al igual que la información de identificación, esta información se proporciona sin una sintaxis especial.

- **Información para recuperar metadatos de la base de datos**

  Esta información se suministra en la mayoría de los módulos de Voai. Este tipo de información son consultas basadas en SQL que permiten recuperar metadatos de una base de datos necesarios para responder a las peticiones OAI-PMH. Estas consultas se

proporcionan con una sintaxis especial, que como ya se había dicho, esta basada en la sintaxis del lenguaje de consulta SQL. A cada consulta que se proporciona en estos módulos le corresponden cuatro campos o parámetros. Estos campos son:

- Correspondencia. Este campo correspondencia es muy importante. Es aquel que permite, como su nombre lo dice, hacer la correspondencia del metadato de la base de datos con el metadato que se va a compartir. Por ejemplo, se hace la correspondencia entre el metadato “titulo” de la base de datos con el metadato “title” del formato Dublín Core que se está compartiendo con el servidor OAI.
- Select. Este campo contiene la sección de la consulta en SQL correspondiente a los atributos que se desean recuperar de la base de datos.
- From. Este campo contiene la sección de la consulta en SQL correspondiente a las tablas que contienen los atributos que se desean recuperar de la base de datos.
- Where. Este campo contiene la sección de la consulta en SQL correspondiente a restricciones que deben cumplir los metadatos que se recuperan de la base de datos.

En ocasiones los metadatos que se desean compartir no se encuentran en la base de datos, sin embargo resulta necesario compartirlos. Ante esa situación, el usuario puede suministrar directamente el valor del metadato en el campo correspondencia, y los demás campos permanecen vacíos pues no se cuenta con una consulta en SQL que permita recuperar el metadato de la base de datos.

Como se puede ver, existen finalmente dos casos en el suministro de información para recuperar metadatos. El primero caso consiste en recuperar metadatos directamente de una base de datos a través de una consulta, y el segundo caso consiste en proporcionar

directamente el valor del metadato en el campo correspondencia debido a que no se cuenta con una consulta capaz de recuperar ese metadato. Para cada caso, la sintaxis a seguir es un tanto distinta. Esta sintaxis se explica a continuación:

- El metadato se recupera de la base de datos

  Para este caso, la información que se proporciona es la consulta que recupera el metadato, es decir, se suministra la información para los campos “correspondencia”, “select”, “from” y “where”. Para explicar la sintaxis en que se proporciona la información en cada campo, se tiene el siguiente ejemplo.

  Hay que suponer que uno de los metadatos que se van a compartir es el metadato “creator” de Dublín Core y que se tiene la información que se muestra en la tabla C.1.

Tabla C.1. Tabla de ejemplo para explicar la sintaxis utilizada en Voai

Tesis		
idTesis	**nombreAutor**	**apellidoAutor**
17	“Abraham”	“Villegas”
23	“Héctor”	“Pérez”
24	“Ricardo”	“Luna”

  Si el usuario desea recuperar, entre otros, al metadato “creator” del registro de la base de datos con identificador 23, la petición OAI-PMH que se haría sobre el servidor OAI es: “verb=GetRecord&metadataPrefix=oai_dc&identifier=23”. Este parámetro “identifier” es el que permitirá saber a que registro corresponderán los

metadatos que se van a recuperar, entonces este parámetro corresponde al atributo "idTesis".

Se puede ver que no existe un metadato "creator" en la tabla "Tesis". Es por eso que es necesario hacer una correspondencia entre los metadatos de la base de datos con los metadatos del estándar de metadatos que se desea compartir. En este caso los atributos "nombreAutor" y "apellidoAutor", ambos, corresponden al metadato "creator".

Por lo tanto, para la petición OAI-PMH ejemplo, se recuperaran los atributos "nombreAutor", "apellidoAutor" correspondientes a "creator" con "idTesis" igual a 23 correspondiente al parámetro "identifier" igual a 23 de la petición OAI-PMH.

La forma de proporcionar información a los cuatro campos de consulta para recuperar los atributos correspondientes a "creator" es la siguiente:

Correspondencia: nombreAutor,$ $,apellidoAutor

Select: *

From: Tesis

Where: idTesis=$identifier$

Se puede ver que en el campo "correspondencia" se tiene el valor "nombreAutor,$ $,apellidoAutor". Como ya se había dicho, los atributos "nombreAutor" y "apellidoAutor" de la base de datos corresponden al metadato "creator", es por eso que aparecen esos atributos como valor del campo "correspondencia". Si se

recuperan y concatenan esos atributos tal y como residen en la base de datos, el valor del metadato “creator” para el “identifier” o “idTesis” 23 sería “HéctorPérez”. Como se puede ver, ese valor no tiene el formato deseado, es decir, uno esperaría que el nombre y el apellido estuvieran separados por un espacio en blanco. Para resolver este problema, se usa el símbolo $. Cualquier texto que se quiera agregar a los atributos recuperados deberá ser delimitado por el símbolo $. Además se deberá escribir una coma entre los atributos que se recuperan de la base y los pedazos de texto delimitados por el símbolo $. Retomando el ejemplo, se puede ver que se tiene el valor “nombreAutor,$ $,apellidoAutor” en el campo “correspondencia”. Este valor nos muestra que se recuperará el atributo “nombreAutor”, posteriormente se le concatenará el texto que se encuentra entre los dos símbolos $, es decir, se le concatenará un espacio en blanco, y finalmente se concatenará el atributo “apellidoAutor”. Por lo tanto, gracias a esta sintaxis, el valor del metadato “creator” para el “identifier” o “idTesis” 23 es “Héctor Pérez”.

El último punto a analizar es el valor suministrado en el campo “where”. Como ya se había dicho, el parámetro “identifier” de la petición OAI-PMH corresponde al identificador de los recursos de la base de datos. En este ejemplo, “identifier” corresponde a “idTesis”. Es por eso que el valor del campo “where” es “idTesis=$identifier$”. De esta manera, el servidor OAI sabrá que el valor del parámetro “identifier” de la petición OAI-PMH será colocado en lugar del texto “$identifier$”. Entonces, para la petición OAI-PMH de ejemplo, la consulta SQL que se construiría dinámicamente para recuperar el metadato “creator” sería:

select nombreAutor,apellidoAutor

from Tesis

where idTesis=23

- El metadato no se puede recuperar de la base de datos

  En caso de que la base de datos no contenga el metadato que se desea compartir, y si el valor de ese metadato es constante, el valor de ese metadato se puede suministrar en el campo “correspondencia” que le corresponde. Por ejemplo, hay que suponer que se desea compartir el metadato “format” de Dublín Core. Si el valor de ese metadato es “application/pdf”, y además es un valor constante para ese metadato, la forma de suministrar la información para el metadato “format” es:

  Correspondencia: application/pdf

  Select:

  From:

  Where:

  Como se puede ver, solo es necesario suministrar el valor del metadato al campo “correspondencia”. Obviamente los otros tres campos permanecen vacíos pues no existe una consulta en SQL capaz de recuperar el metadato “format”.

Ahora que ya se sabe la sintaxis utilizada para suministrar las consultas en SQL que recuperan los metadatos a compartir. Será mucho más fácil explicar la forma en que se proporciona información para cada módulo de especificación de la fase 1 de Voai.

C.2.2 Módulo de especificación correspondiente a GetRecord

En este módulo se proporciona una consulta que permita recuperar un registro específico de la base de datos. Por ejemplo, si se tiene la información de la tabla C.2, la consulta que se proporcionaría en los campos de este módulo se puede ver en la interfaz de la figura C.3.

Tabla C.2. Tabla de ejemplo para explicar el módulo GetRecord

tesis		
idTesis	**tituloTesis**	**autorTesis**
“is114”	“titulo 114”	“autor 114”
“is117”	“titulo 117”	“autor 117”



Figura C.3. Interfaz del módulo correspondiente a GetRecord

Entonces, tal y como se había explicado en la sección C.2.1 de este documento, la parte del texto correspondiente a "$identifier$" que se encuentra como valor del campo "where" será sustituida por el valor del parámetro "identifier" de las peticiones OAI-PMH.

C.2.3 Módulo de especificación correspondiente a Identify

En este módulo se específica información de identificación de la colección. Esta información es:

- Nombre del repositorio. En este campo se dará el nombre que se desea tenga el servidor OAI generado, generalmente ese nombre es el nombre de la colección o base de datos.
- Url base: En este campo se proporciona un url formado por el host, puerto y servlet principal que escucha las peticiones OAI-PMH. Voai generará e instalará el servidor OAI de tal manera que podrá se accedido en el base url: http://localhost:8080/ServidorOAI/Oai_handler
- Correo electrónico del administrador. Aquí se proporciona el correo electrónico de la persona que se encargará de la administración, evaluación y registro del servidor OAI.
- Fecha más antigua de modificación: En este campo se proporciona la fecha más antigua en que fue guardado o modificado un registro de la base de datos. Por ejemplo 1995-07-28
- Registros eliminados: en este campo se especifica si la base de datos mantiene información sobre los registros borrados. Los posibles valores son:
    - no: la base no mantiene información sobre los registros borrados
    - transient: la base de datos si mantiene la información de registros borrados

  - persistent: no se garantiza que se mantenga toda la información sobre registros borrados

- Granularity: en este campo se especifica la granularidad en que se mantienen las fechas de registro en la base de datos. Esta granularidad puede ser YYYY-MM-DD o YYYY-MM-DDThh:mm:ssZ.
- Identificador del repositorio: Se especifica como se desea que sea el identificador del servidor OAI de acuerdo al patrón [a-zA-Z][a-zA-Z0-9\-]*(\.[a-zA-Z][a-zA-Z0-9\-]+)+. Ejemplo: incunables.uva
- Ejemplo de identificador: aquí se da un ejemplo de como sería un identificador real del servidor OAI que será generado. Por ejemplo: oai:incunables.uva:114

La interfaz de este módulo de configuración se muestra en la figura C.4 de este documento.

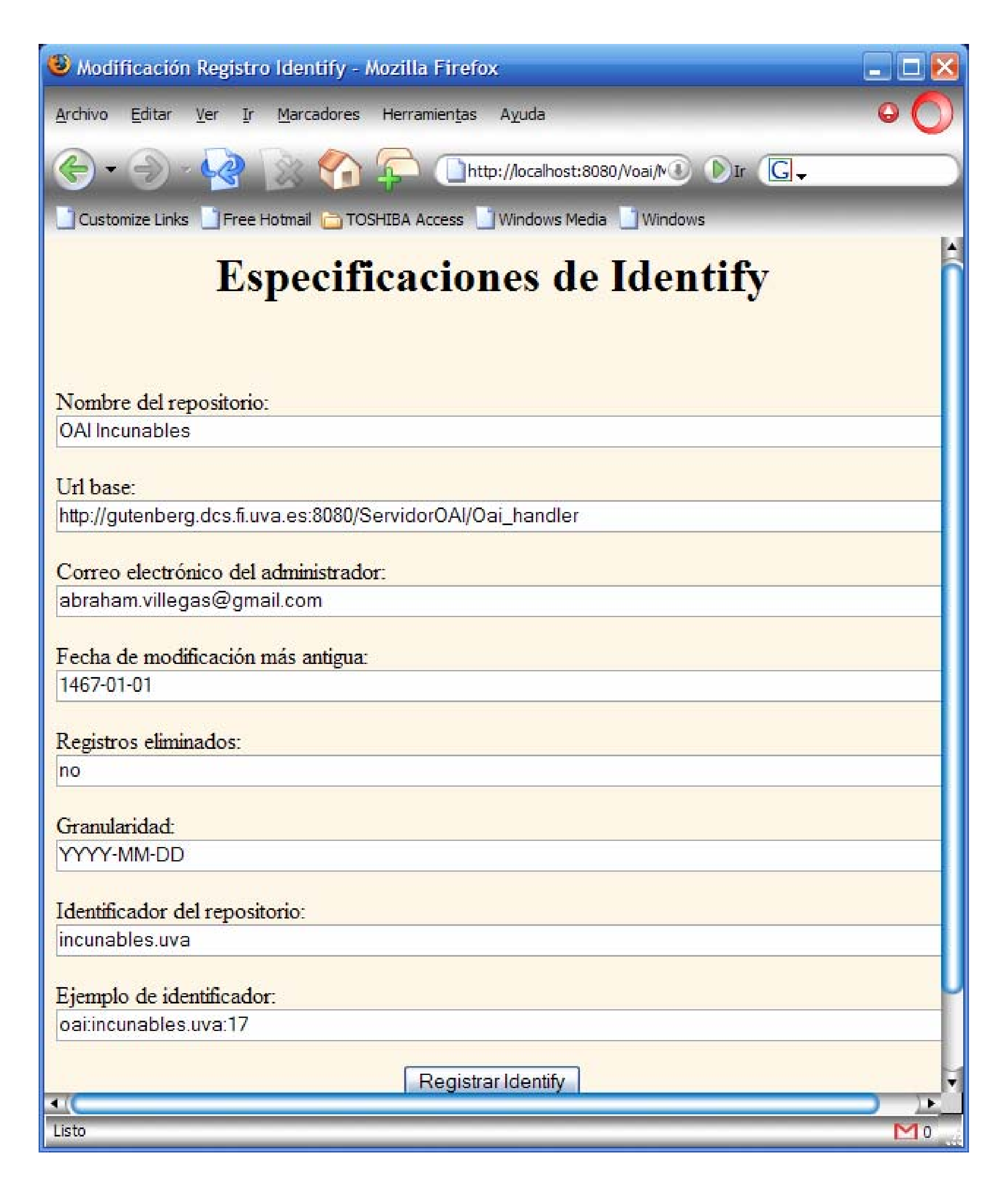


Figura C.4. Interfaz del módulo correspondiente a Identify

C.2.4 Módulo de especificación correspondiente a ListIdentifiers y ListRecords

En este módulo se especifican dos consultas (en caso de que la colección soporte el manejo de conjuntos). Las consultas son:

- Se proporciona la consulta que permite recuperar los conjuntos a los que pertenece un registro específico.

Tabla C.3. Tabla de ejemplo para explicar el módulo ListRecords y ListIdentifiers

tesis		
idTesis	**autorTesis**	**departamento**
“is111”	“autor 111”	“sistemas”
“ad087”	“autor 087”	“administración”

Por ejemplo, si se tiene la información de la tabla C.3, y si a partir de esta información se considera que los conjuntos que soporta esta base de datos son los departamentos, entonces la consulta que se proporcionaría es la siguiente:

Correspondencia: departamento

Select: *

From: tesis

Where: idTesis=$identifier$

Si, por ejemplo, en la petición OAI-PMH el valor del parámetro “identifier” es “is111”, la consulta nos proporcionaría los conjuntos que soporta ese identificador, en este caso es solo uno: “sistemas”.

- La segunda consulta debe permitir recuperar los registros que pertenecen a un conjunto específico, es decir, es la consulta contraria a la vista en el punto anterior. Entonces, tomando nuevamente como ejemplo la información de la tabla C.3, la consulta sería:

  Correspondencia: idTesis

  Select: *

  From: tesis

  Where: departamento=set

  Si, por ejemplo, en la petición OAI-PMH se solicitan los registros que pertenecen al conjunto "sistemas", la petición sería "verb=ListRecords&metadataPrefix=oai_dc&set=sistemas". Al igual que se hace la correspondencia del parámetro "identifier" con el metadato que identifica de manera única a un recurso de la base de datos, también se hace una correspondencia entre el parámetro "set" de la petición OAI-PMH y el atributo correspondiente a los conjuntos que soporta la base de datos. En este ejemplo, se hace la correspondencia de "set" con "departamento". Entonces, retomando la petición OAI-PMH, la consulta proporcionada en el módulo de especificación me proporcionaría los registros del conjunto "sistemas", y en este caso el único registro resultante es "is111".

La interfaz de este módulo de especificación se muestra en la figura C.5 de este documento.

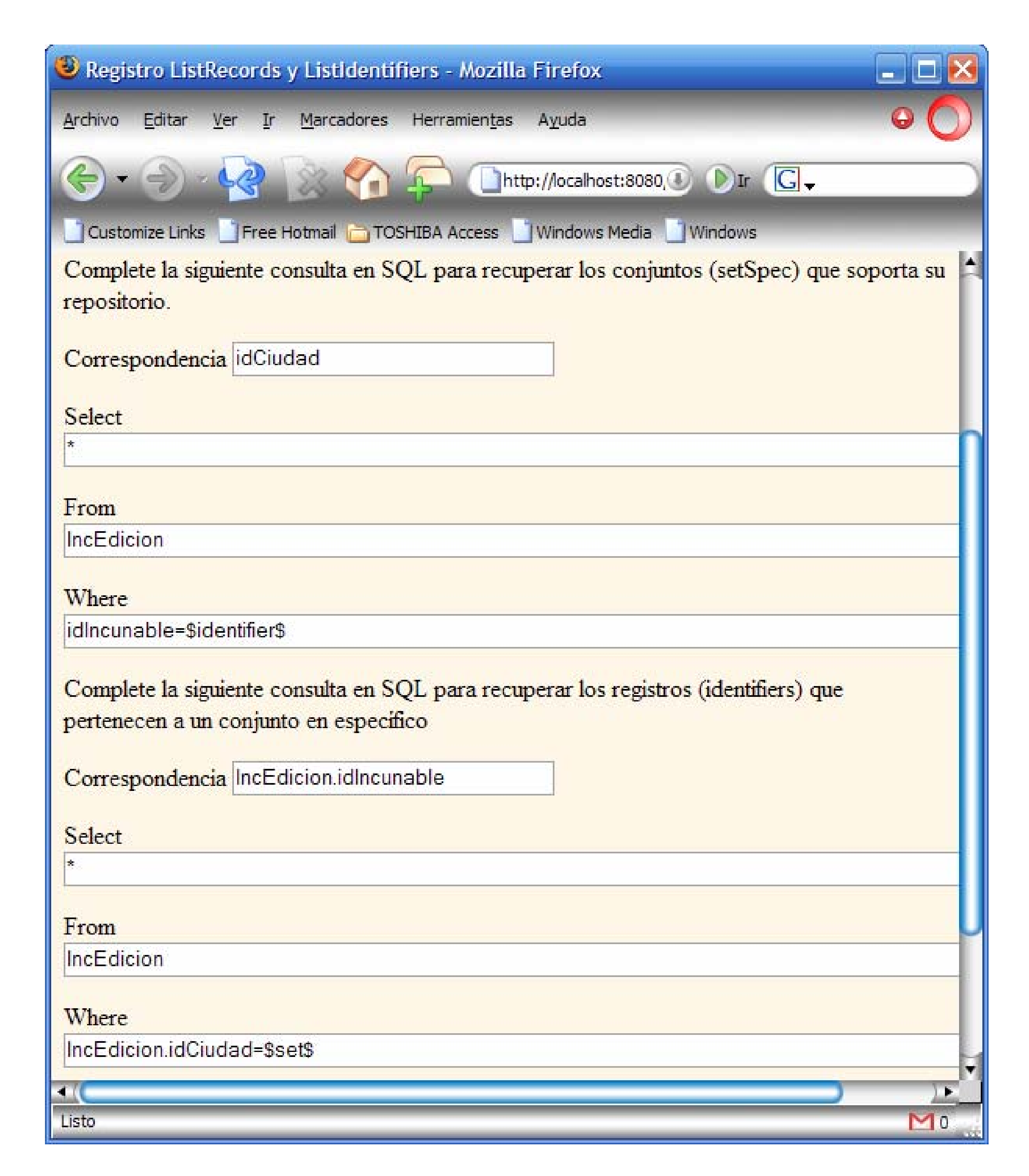


Figura C.5. Interfaz del módulo correspondiente a ListRecords y ListIdentifiers

C.2.5 Módulo de especificación correspondiente a ListMetadataFormats

En este módulo se especifican las consultas para recuperar los metadatos para los estándares de metadatos que soportará el servidor OAI. Este módulo se divide en dos:

- Especificación de consultas para el estándar de metadatos Dublín Core.

  Dublín Core es el estándar de metadatos mínimo que debe soportar un servidor OAI debido a la generalidad de sus metadatos. Para este estándar de metadatos es necesario especificar:

  - El prefijo del formato Dublín Core
  - La ubicación del esquema de validación de Dublín Core
  - La ubicación del "nameSpace" de Dublín Core
  - El elemento inicial del documento xml que soporta Dublín Core.
  - Para cada metadato de Dublín Core que se soporte, se especifica una consulta que pueda recuperar ese metadato de la base de datos, o bien, se especifica un valor constante para ese metadato. Para esto, en caso de que la información para un metadato de Dublín Core sea una consulta, se suministra información a los cuatro campos (correspondencia, select, from, where). En caso de que el valor de ese metadato no se pueda recuperar de la base de datos, se suministra un valor constante al campo "correspondencia". Los detalles de la forma en que se proporciona información se muestran en la sección C.2.1 de este documento.

Tabla C.4. Tabla de ejemplo para explicar el módulo ListMetadataFormats

tesis		
idTesis	**autorTesis**	**tituloTesis**
"is111"	"autor is111"	"titulo is111"
"ad087"	"autor ad087"	"titulo ad087"

Por ejemplo, si se tiene la información de la tabla C.4, y se desea compartir el metadato "creator" de Dublín Core, se hace la siguiente consulta:

Correspondencia: autorTesis

Select: *

From: tesis

Where: idTesis=$identifier$

Gracias al campo correspondencia, el generador sabe que el metadato "autorTesis" corresponde al metadato "creator" de Dublín Core. Este mismo proceso de especificación de consultas se hace para los demás metadatos de Dublín Core. La interfaz del módulo de configuración para ListMetadataFormats se puede ver en la figura C.6. Además, en esta figura se ve el suministro de información correspondiente a los archivos de validación del estándar Dublín Core, como la consulta correspondiente al metadato "Title" de este mismo formato de metadatos.

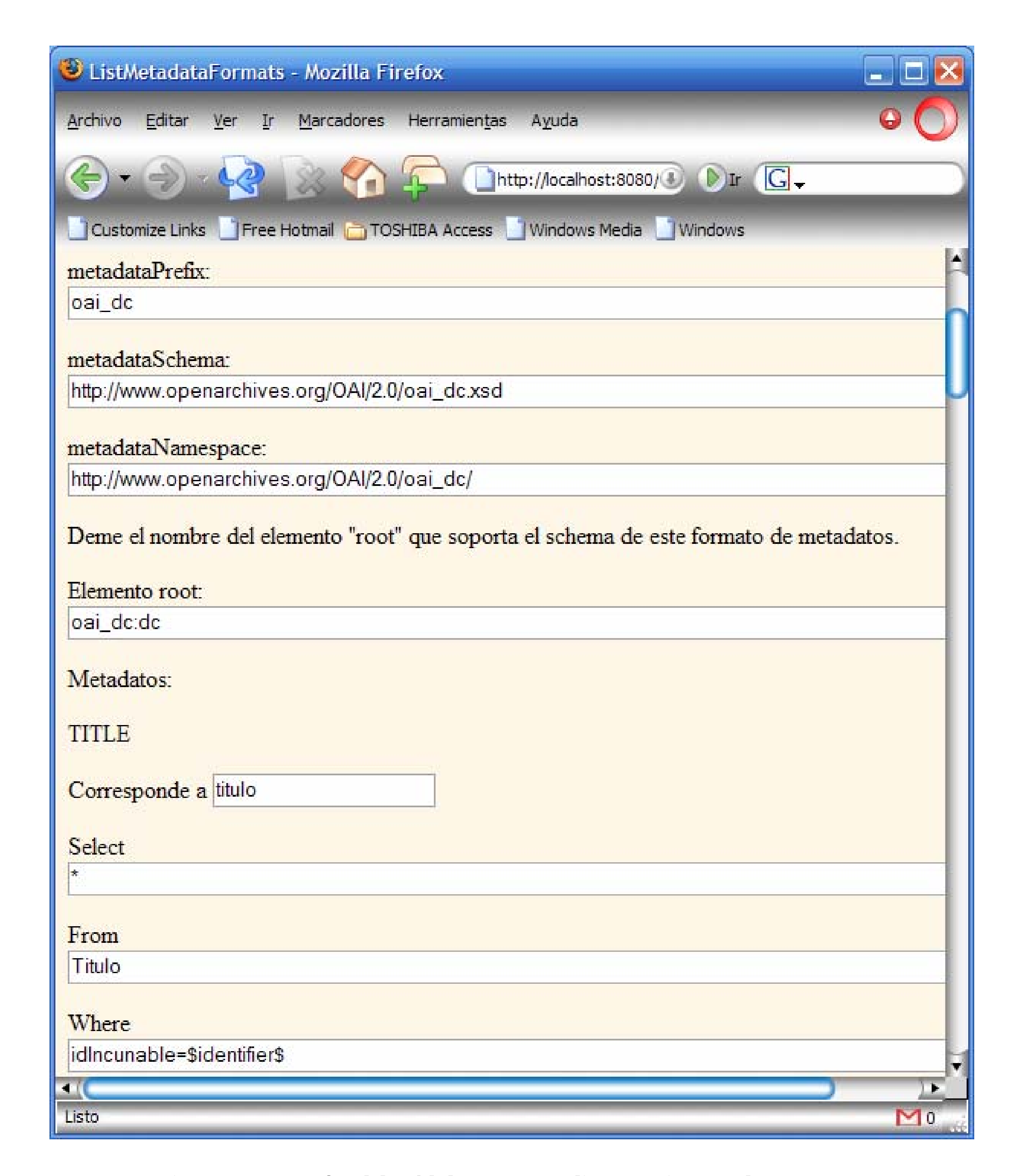


Figura C.6. Interfaz del módulo correspondiente a ListMetadataFormats

- Especificación para otros estándares de metadatos.

  Si se desea compartir los metadatos de la base de datos con otros estándares distintos a Dublín Core, el proceso a seguir es el siguiente:

- Al final de los campos correspondientes a los metadatos de Dublín Core, se selecciona la opción de que se desea soportar otro formato de metadatos y se proporciona el número de metadatos que tiene ese formato. Este paso se puede ver en la figura C.7 de este documento.
- Después de especificar el número de metadatos para el nuevo estándar, aparece una nueva interfaz en donde se suministra la información para cada metadato de ese estándar. Esta interfaz se puede ver en la figura C.8. Algo importante en este punto es que para cada metadato, además de los cuatro campos estándar (correspondencia, select, from, where), existe un campo nuevo llamado "nombre del metadato". En este campo, como su nombre lo dice, se proporciona el nombre de ese metadato para el estándar que se está especificando.
- Al terminar la especificación de un estándar, se puede continuar creando nuevos estándares de metadatos, o se puede terminar con este proceso.

Por ejemplo, hay que suponer que se desea otro formato de metadatos, llamémosle "oai_prueba", con solo dos metadatos que son "tituloPrueba" y "autorPrueba". Retomando la información de la tabla C.4 de este documento podemos hacer la correspondencia entre los metadatos de la base y los metadatos del estándar que se va a compartir. Si el metadato "tituloTesis" corresponde a "tituloPrueba" y el metadato "autorTesis" corresponde a "autorPrueba", la especificación del nuevo formato se hace de la siguiente manera:

- Se especifica en la interfaz de ListMetadataFormats que se soportará otro formato con dos metadatos. Esta interfaz se puede ver en la figura C.7.

- Se especifica un prefijo (oai_prueba), la ubicación del esquema, el nameSpace y el nombre del elemento inicial del documento xml soportado por el esquema.
- Se especifica la información para el metadato “autorPrueba” de la siguiente manera:

  Nombre del metadato: autorPrueba

  Correspondencia: autorTesis

  Select: *

  From: tesis

  Where: idTesis=$identifier$
- Se especifica la información para el elemento “tituloPrueba” de la siguiente manera:

  Nombre del metadato: tituloPrueba

  Correspondencia: tituloTesis

  Select: *

  From: tesis

  Where: idTesis=$identifier$

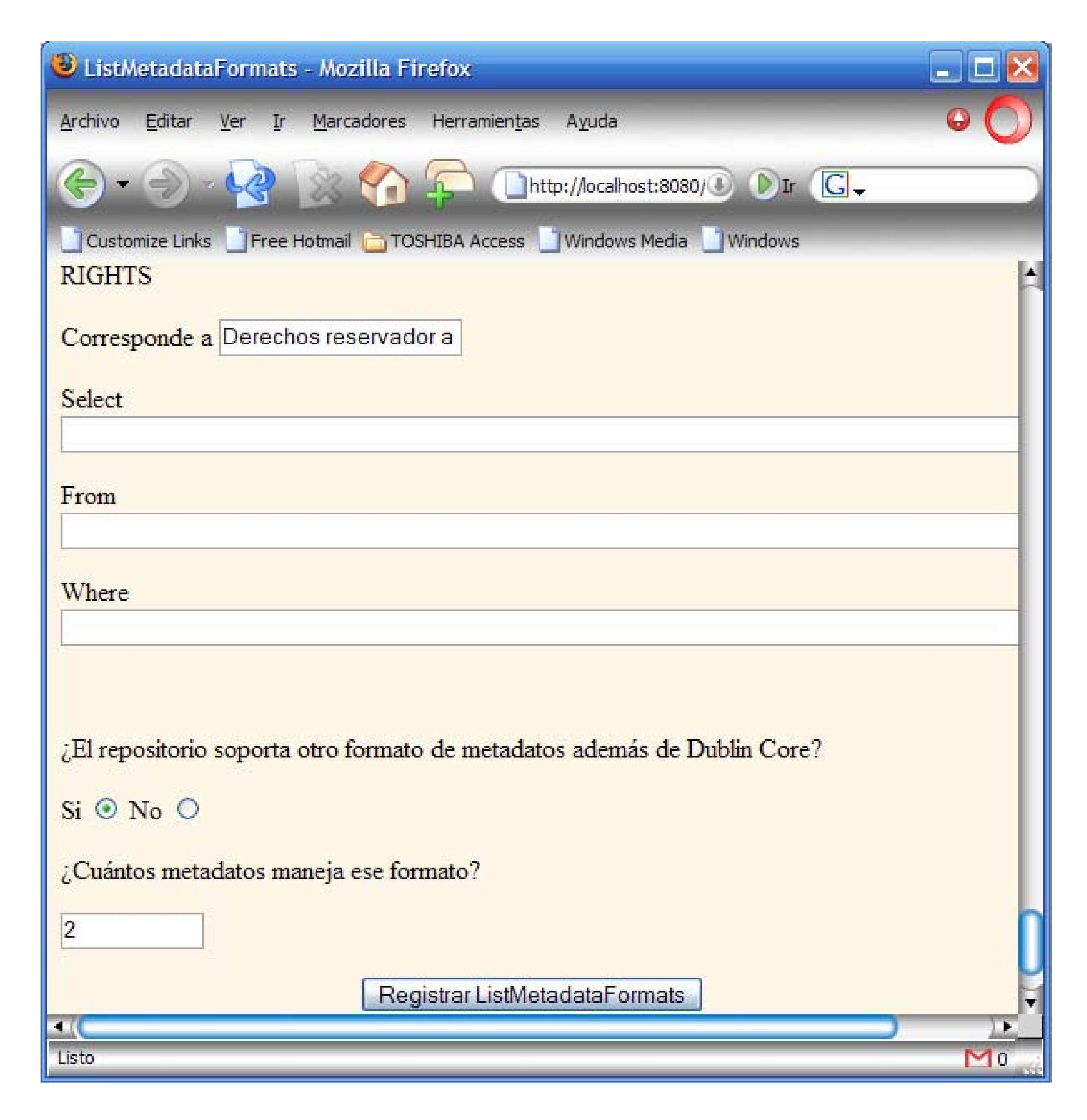


Figura C.7. Creación de un nuevo estándar de metadatos

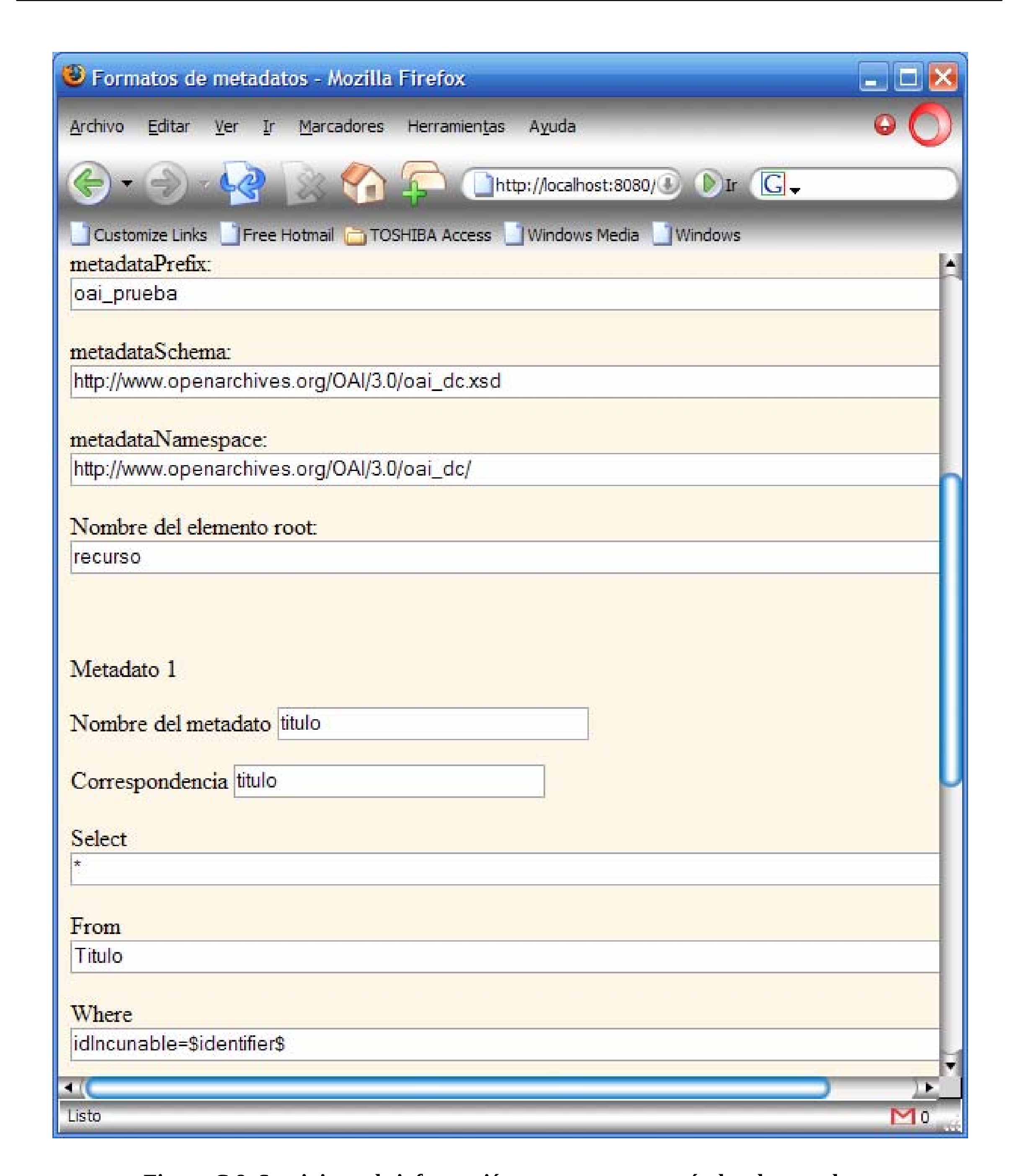


Figura C.8. Suministro de información para un nuevo estándar de metadatos

C.2.6 Módulo de especificación correspondiente a ListSets

En este módulo se especifica tanto información referente a los conjuntos que soporta la base de datos, como los parámetros de conexión a la base de datos. La interfaz de este módulo se puede ver en la figura C.9 de este documento.

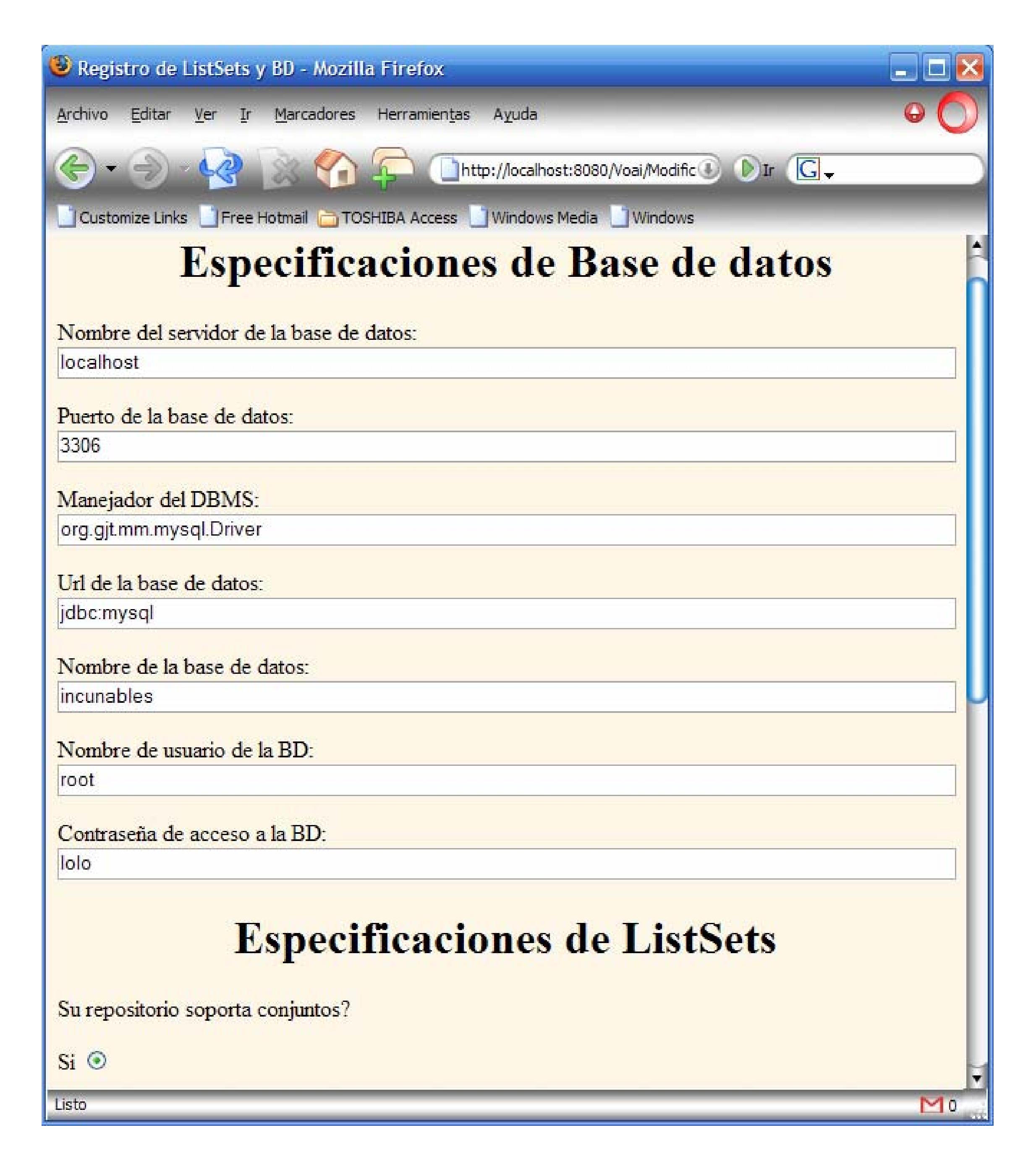


Figura C.9. Interfaz del módulo correspondiente a ListSets

Para este módulo, en caso de que se soporten conjuntos, se necesita proporcionar una consulta en SQL que permita recuperar los nombres de los conjuntos que soporta la base (SetName) y sus identificadores (SetSpec).

Por ejemplo, si se tiene la información de la tabla C.5, y los conjuntos que soporta la base son los departamentos, la consulta se proporcionaría de la siguiente manera:

Select: *

From: departamentos

Where:

Y en los campos de correspondencia la información que se proporcionaría es:

SetSpec: idDepartamento

SetName: nombreDepartamento

Tabla C.5. Tabla de ejemplo para explicar el módulo ListSets

departamentos	
idDepartamento	**nombreDepartamento**
“dis”	“depto. Sistemas”
“dad”	“depto. Administración”

C.3. Fase dos: selección de ubicación donde se instalará el servidor OAI

Después de haber terminado con el proceso de suministro de información para cada módulo de especificación de la fase 1, se procede con la fase 2 de Voai. Para acceder a esta fase 2, se selecciona el segundo hipervínculo de la interfaz principal de Voai. Esta segunda fase es muy simple. Consiste en seleccionar la ruta en donde se desea que se instale el servidor OAI que se va a generar. La interfaz de esta fase se muestra en la figura C.10.

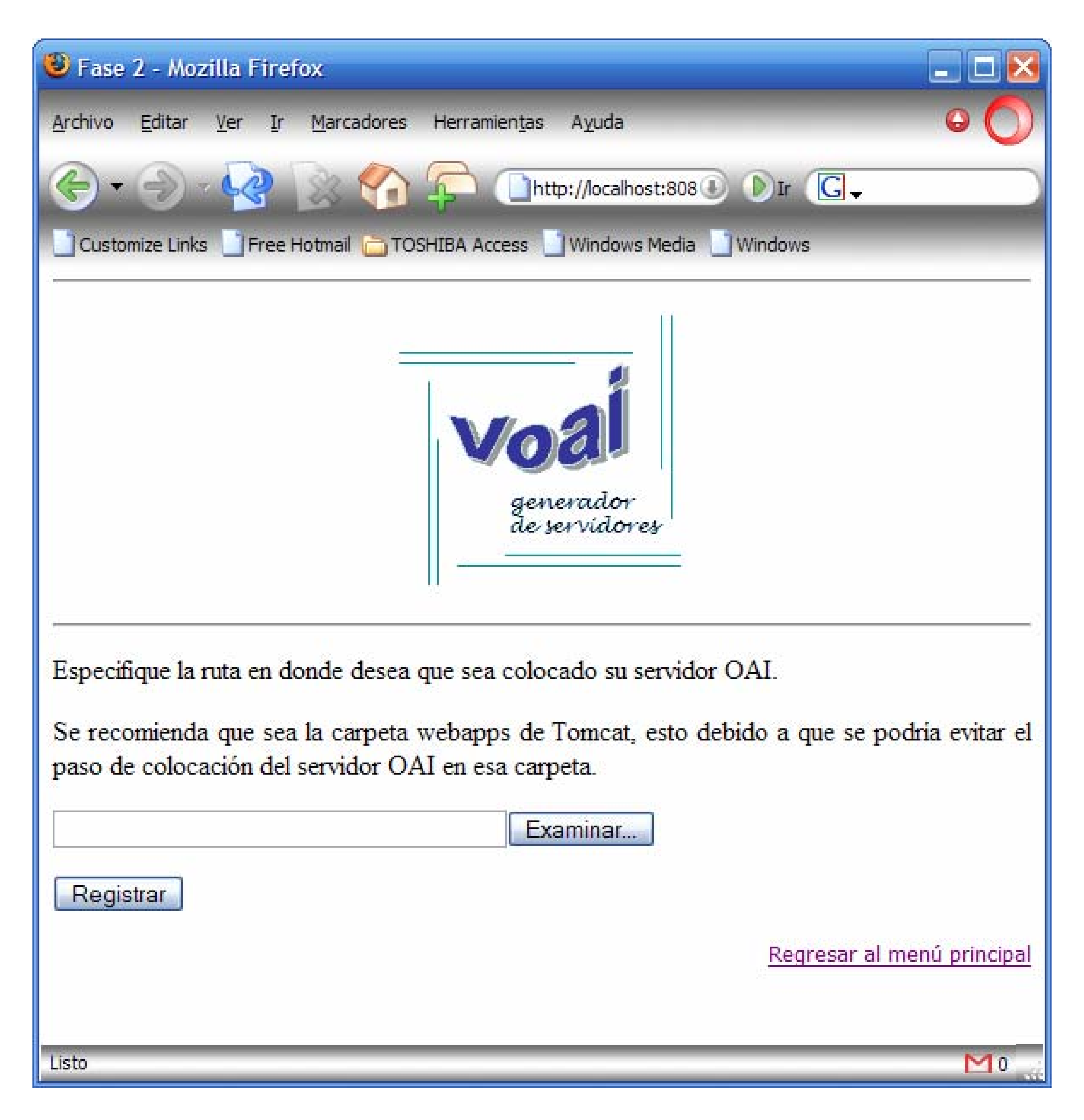


Figura C.10. Interfaz de la fase 2.

C.4. Fase tres: generación, instalación y ejecución del servidor OAI

En esta última fase se hace la generación, instalación y ejecución del servidor OAI. Sin embargo, los pasos a seguir son muy sencillos. La interfaz de esta fase tres se puede ver en la figura C.11 de este documento. Para acceder a la fase 3 se selecciona el tercer hipervínculo de la interfaz principal de Voai, y se siguen los siguientes cinco pasos:

- Se detiene la ejecución de Tomcat. Para esto se utiliza la herramienta “Monitor Tomcat” (aplicación que instala Tomcat), o bien se ejecuta el archivo “shutdown.bat” de la carpeta “bin” de Tomcat.
- Se ejecuta el archivo “Install.bat” que se encuentra en la ruta de instalación seleccionada en la fase 2. Este archivo es que creará la estructura de archivos necesaria que conformará al servidor OAI.
- Se coloca la carpeta “ServidorOAI” generada en la carpeta “webapps” de Tomcat.
- Se inicia la ejecución de Tomcat. Para esto, nuevamente se hace uso de “Monitor Tomcat”, o bien se ejecuta el archivo “startup.bat” de la carpeta “bin” de Tomcat.
- En este momento ya está montado en Tomcat el servidor OAI generado. Para empezar a utilizar el servidor OAI, acceder a la dirección http://localhost:8080/ServidorOAI.

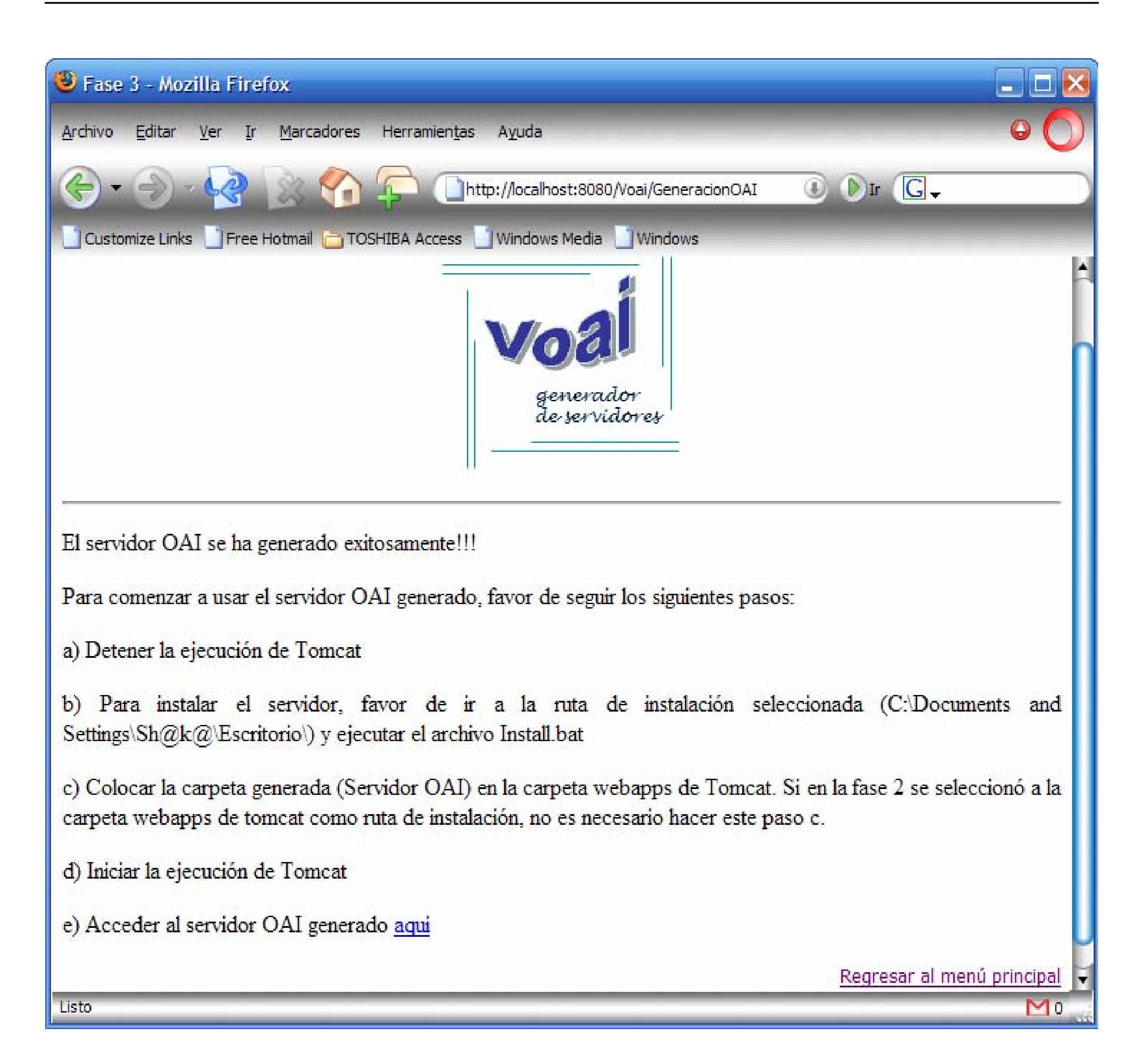


Figura C.11. Interfaz de la fase 3.